I.J. Image, Graphics and Signal Processing, **2025, 5, 26-41**
Published Online on October 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2025.05.03

Classification of Soil Images Using Convolutional Neural Network

Girish D. Chate*
School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra-425001, India
E-mail: chategirish@gmail.com
ORCID iD: https://orcid.org/0009-0002-4081-4060
*Corresponding Author

S. S. Bhamare
School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon, Maharashtra-425001, India
E-mail: ssbhamare.nmu@gmail.com
ORCID iD: https://orcid.org/0000-0002-6160-8533

Received: 11 May, 2024; Revised: 04 June, 2025; Accepted: 14 July, 2025; Published: 08 October, 2025

Abstract: Soil image classification plays a crucial role in agricultural and environmental practices. Traditional methods of soil classification often involve manual labor, which can be time-consuming and prone to human error. Recent advances in computer vision and machine learning have opened new horizons for automating this classification process. This research paper presents a comprehensive study and evaluates the performance of four convolutional neural network (CNN) architectures a custom CNN, ResNet50, InceptionV3, and MobileNetV2 on a custom soil image dataset comprising 1800 labelled images across four soil classes such as Black, Laterite, Red and White. The dataset created using smartphone camera to captured images under varying natural conditions. The objective of this work is to explore the effectiveness and accuracy of different machine learning algorithms used in categorizing soil types based on visual data. Each model's performance is evaluated in terms of classification accuracy, precision, recall, and F1-score. Results indicate that ResNet50 achieves the highest accuracy 97.3%, followed closely by MobileNetV2 94.7%. The custom CNN, while computationally efficient, achieved 88.2%. We conclude that transfer learning with deep CNNs is highly effective for soil classification, and MobileNetV2 is a strong recommended for mobile applications. The comparative analysis demonstrates their effectiveness in distinguishing between different soil types, textures, and compositions. It also highlights how important it is to select the appropriate CNN architectures for certain tasks related to soil classification. This work belongs to the increasing collection of information at the interface between soil science and computer vision. It offers a strategy to apply sophisticated deep learning-based algorithms to assess soil type more reliably and effectively, serving as a springboard for future research in the field of soil image analysis and classification.

Index Terms: Agriculture, Soil Image, Soil Classification, CNN, ResNet50, InceptionV3, MobileNetV2, Machine Learning

1. Introduction

Soil type classification plays a critical role in modern agriculture, influencing key decisions related to crop selection, fertilizer recommendations, and irrigation management. Accurate knowledge of soil properties such as texture, colour, and structure is essential for optimizing crop yield and ensuring sustainable land use. Traditional soil classification methods, such as laboratory testing and field sampling, are often time consuming, expensive, and labour-intensive, requiring expert knowledge and sophisticated equipment [1]. With the rapid advancement in computer vision and deep learning, particularly Convolutional Neural Networks (CNNs), there has been increasing interest in using image-based approaches for automated soil classification. CNNs can extract high-level spatial and colour features from images, making them suitable for tasks like object detection, segmentation, and classification [2]. This enables the development of scalable systems that can classify soil types directly from images taken by smartphones or drones, providing a low-cost and accessible solution for farmers and agronomists in remote and resource-constrained areas [3].

This paper demonstrated the potential of CNNs for soil image classification, using both custom and public datasets [4]. Pre-trained models such as ResNet50, InceptionV3, and MobileNetV2 have been widely adopted due to their demonstrated accuracy and robustness in various visual recognition tasks. ResNet50, known for its residual connections, allows for deeper network training without vanishing gradients [5]. InceptionV3 employs multi-scale convolution filters to capture complex image features, while MobileNetV2 offers a lightweight architecture ideal for mobile and embedded devices [6,7].

However, despite the growing body of research, limited work has focused on comparing these architectures using real-world, custom soil datasets that reflect diverse soil conditions and lighting variations encountered in the field. This paper aims to fill that gap by conducting a comparative evaluation of CNN, ResNet50, InceptionV3, and MobileNetV2 architectures on a custom soil image dataset. The objective is to determine which architecture offers the best trade-off between classification performance and computational efficiency for potential deployment in mobile-based smart agriculture applications.

2. Related Work

Many researchers have worked in this area of image-based soil classification and most of them focus on identification of soil properties like physical and chemical based on soil images. The application of deep learning techniques in the domain of soil classification has gained significant momentum over the last decade. Traditional approaches relied on spectral data, physical testing, or expert analysis, which are time-consuming and often impractical for large-scale or real-time implementation. The emergence of Convolutional Neural Networks (CNNs) has enabled the development of automated and scalable soil classification systems based on visual information. Several researchers have explored the potential of CNNs for soil image classification.

Singh et al. [8] developed a CNN-based model to classify soil textures using RGB images, reporting high accuracy on benchmark datasets. Similarly, Ahmad et al. [9] proposed a deep CNN architecture for soil type recognition using UAV-captured images, achieving considerable improvement over traditional machine learning methods like Support Vector Machines (SVM) and Random Forests.

Several studies have used transfer learning to enhance classification performance, especially when dealing with small datasets. Pre-trained models like ResNet50, InceptionV3, and VGG16 trained on large-scale datasets such as ImageNet are fine-tuned for soil classification tasks. For instance, Pujari et al. [10] used ResNet50 for soil fertility classification and demonstrated its superior performance over shallow architectures. In another study, Hossain et al. [11] employed InceptionV3 and achieved higher accuracy due to its ability to capture multi-scale features and complex textures from soil surface images.

Mobile-friendly models like MobileNetV2 have also been applied in recent works, given their computational efficiency and suitability for on-field deployment. Maji and Chakraborty [12] implemented MobileNetV2 on smartphone-captured soil images and showed that it could deliver competitive accuracy while significantly reducing inference time and resource usage, making it ideal for real-time soil monitoring.

Barman U and Choudhury RD [15] introduces a soil classification model that used multiple Support Vector Machine and Linear Kernel functions. The images were captured by a smartphone camera. The Support Vector Machine can be used with a variety of class sizes. Except for loamy sand, silty clay, and loamy fine sand are the three types. The classifier performed on the dataset.

A survey of methods for classifying soil using machine learning was presented by researchers Chandan and Thakur R [16]. Arid soil, peaty soil, mountain soil, laterite soil, snowfields, forest soil, alluvial soil, black soil, red soil, and salty soil are the primary soil types found in India. To categorize soils according to their physical and chemical characteristics, a variety of strategies are employed in the devised systems, such as K-Nearest Neighbor Networks, Decision Trees, Support Vector Machines, and Artificial Neural Networks.

Aziz MM, Ahmed DR et al. [17] proposed techniques for determining soil pH value using Artificial Neural Networks. They have built a database. To evaluate the pH values, the red, green, and blue values of the soil image sample and minimum error are determined by comparing these samples recorded in the database. In this objective of work, three neurons make up the input layer and ten neurons make up the hidden layer, and one node makes up the output layer. The input layers of three input neurons feed three different colors of soil such as red, green, and blue.

The researchers Barman U, Choudhury RD et al. [18] offered a system for determining the pH value of soil without human involvement. Image enhancement techniques such as image scaling, image filtering, and contrast improvement are utilized to decrease the undesired distortions in the soil images. The mean values of Hue Saturation Value components are calculated to provide better results for soil pH values. Best tillage conditions for varying aggregate dimensions according to seed requirements.

Ajdadi FR, Gilandeh YA, et al. [19] suggested methods for employing artificial neural networks to analyse tillage quality in real time. Using methods like the grey level matrix, textural details were taken out of photos of tilled soil.

A soil texture classification method that employs wavelet transform techniques to distinguish between different textures was suggested by Zhang X, Younan NH, et al. [20]. The wavelet transform technique, which is a potent image and signal analysis tool because of its multiple resolution qualities, is used to extract these features.

The best results are obtained using this ML parameter estimate approach. Clay, sand, and silt are examples of soil textures used for training and categorization. A method for using wavelet statistics to categorize soil textures. These two model's maximum likelihood and hidden Markov models are used to create a new system.

Chung SO, Cho KH et al. [21] examined a technique for classification of soil texture constructed on Red (R), Green (G), and Blue (B) form images. Four surface images are taken with a digital camera for each segmented soil sample.

Using a morphological incomplete differential equation-based method based on image contrast, Sofou A, Evangelopoulos G, et al. [22] developed a computer-based texture analysis on soil images for segmentation. To assess surface texture variations, it was suggested to employ variations in soil images as a local modulation component.

To extract textural information for retrieval, Honawad PSK, Chinchali PSS, et al. [23] suggested a soil classification method based on the Gabor Filter on original soil pictures. In a database of 100 soil photos from ten different kinds with varying translation, scales, and orientations, our method showed good retrieval performance.

To establish the quantitative assignment of soil main constituents, such as sand, clay, and silt, using Digital Image Processing and multivariate calibration, de O. Morais PA et al. [24] presented environmentally friendly MIA-based analytical methods for characterizing soil texture using an application for the computer.

Maniyath SR et al. [25] researchers developed an efficient technique for detecting soil color. Hue, Saturation, and Value (HSV) images (B) are created by converting red, green, and blue images of soil. There are lower and upper bounds for the hue component. In this component, the soil image is transformed back into red, green, and blue form. The KNN algorithm uses a distance metric to determine the closest example. Distance metrics such as Euclidean distance are used here, but other distance metrics can be used.

Researchers Gurubasava, Mahantesh SD [26] have suggested picture processing to measure and assess the pH of agricultural soil. pH can be described using concepts like acidity or basicity. The basicity or acidity of the soil affects plant growth. The suggested methods use Principal Component Analysis (PCA) to calculate mean values and extract important features like Red (R), Green (G), and Blue (B) index values. While trained images have 100% accuracy, untrained images have 91% accuracy.

3. Flow Diagram of Proposed Framework

Figure 1 shows the dataset preparation, preprocessing pipeline, and the deep learning architecture evaluated in this study. The proposed framework was designed to enable a fair and systematic comparison between a custom CNN and three popular pre-trained models ResNet50, InceptionV3, and MobileNetV2 using the same soil image dataset. This framework consists of different phases as follows.

A. Raw Soil Images Dataset

This is the input stage where you collect a dataset of soil images. These images are typically taken using smartphone camera in field conditions and are labelled according to soil type, class content.

B. Preprocessing & Data Augmentation

Before feeding images into the model, they undergo to image preprocessing for resizing, normalization, and colour space conversion such as RGB to grayscale. After that data augmentation techniques such as rotation, flipping, zooming, and cropping are applied to artificially increase the diversity and size of the training dataset, helping to improve model generalization and reduce overfitting.

C. Model Selection

Here, the architecture for classification is chosen. Custom CNN a convolutional neural network designed from scratch and pre-trained model's architectures such as ResNet50, InceptionV3, and MobileNetV2 are used with transfer learning. These models are already trained on large datasets like ImageNet and offer good generalization.

D. Feature Extraction

For pre-trained models, feature extraction involves freezing the convolutional layers so that only the classification head is trained. Custom CNNs, all layers are trainable, allowing the model to learn features from scratch.

E. Training (Compile + Fit Model)

In this stage the model is **compiled** with a loss function (categorical cross-entropy), optimizer (Adam), and metrics (accuracy). The model is then **trained** on the pre-processed and augmented dataset over ten epochs using batch learning.

F. *Classification Head (GlobalAvgPooling + Dense)*

The final layer of the model is global average pooling reduces each feature map to a single value, reducing the total number of parameters and preventing overfitting and dense layers act as the classifier, producing output probabilities for each soil class using a SoftMax activation function.

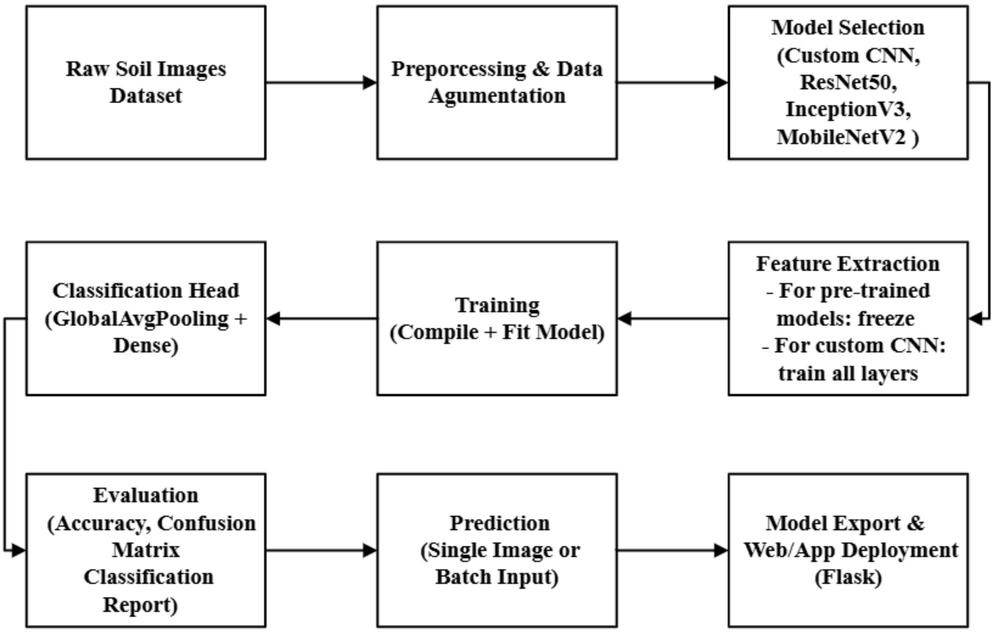


Fig. 1. Flow Diagram of Proposed Framework

G. *Evaluation*

After training, the model's performance is assessed using Accuracy for overall correctness, confusion matrix for to see how many samples are correctly and incorrectly classified in each class, and classification report to includes precision, recall, and F1-score per class.

3.1 Dataset of Soil Images

A custom soil image dataset was developed for this study. Soil samples collect four major types of Black, Laterite, Red, and White these are collected from various agricultural fields across multiple geographic regions. Each soil sample photo capture under varying lighting conditions, orientations, and backgrounds using smartphone cameras. After assign nine class label for four major soil type such as Black soil (B1, B2, B3), Laterite soil (L1, L2), Red soil (R1, R2) and White soil (W1, W2). The dataset consists of 1800 labelled images, with 200 samples per class, and aims to capture real-world diversity in soil colour, texture, and composition.

Table 1. Labels and Categories dataset of Soil Images

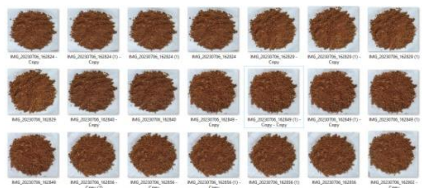
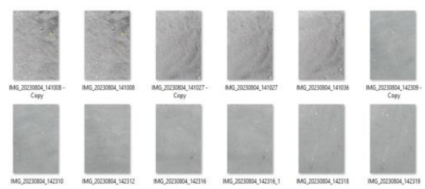
Categories / Labels	**Dataset of Soil Images**	**Labels / Categories**	**Dataset of Soil Images**
Black Soil (B1, B2, B3)		Laterite Soil (L1, L2)	
Red Soil (R1, R2)		White Soil (W1, W2)	

Figure 2 shows the class-wise distribution of soil images within the dataset, comprising nine distinct categories such that B1, B2, B3, L1, L2, R1, R2, W1, and W2. Each category contains an equal number of 200 images, resulting in a total of 1800 samples. This uniform distribution ensures class balance, which is a critical factor in training deep learning models for classification tasks[13].

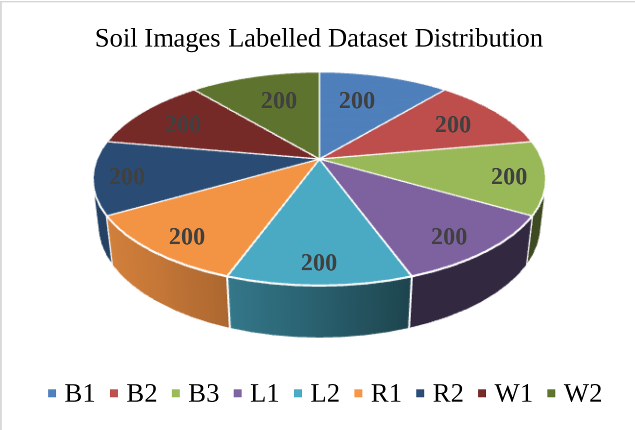


Fig. 2. Distribution of Soil Images Labelled Dataset

Images were captured at a resolution of 1024x768 pixels and then resized to 224x224 pixels to match the input requirements of the pre-trained CNN architectures. To improve the robustness and generalization of the models, data augmentation techniques such as random rotation, horizontal flipping, vertical flipping, brightness adjustment, and zooming are applied during training [14].

3.2 Image Preprocessing

In this task, computer vision, image analysis and essential image preparation are involved. Before supplying the raw input images to a machine learning model, a variety of procedures must be used. Preprocessing images have multiple uses that improve the model's overall efficacy and efficiency. Figure 3 shows steps of image processing.

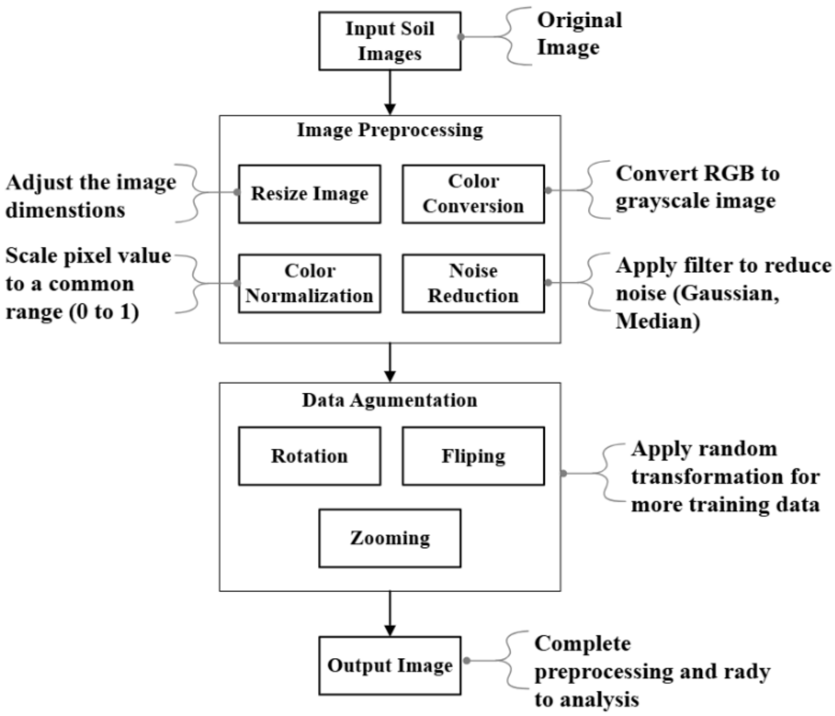


Fig. 3. Flow Diagram of Image Preprocessing

3.2.1 Input Soil Images: This is the raw data, consisting of original soil images collected from the field or laboratory. These images are likely in RGB format and may vary in size, lighting, and quality.

3.2.2 Image Preprocessing: This step prepares the images for analysis and improves the performance of the classification models.

a. Resize Image: Adjust all input images to a uniform size 224x224 pixels to ensure compatibility with CNN model input layers these are neural networks require fixed input dimensions.

b. Colour Conversion: Convert RGB images to grayscale if needed these are reduces computational complexity or focuses on texture features instead of colour when colour is not informative.

c. Colour Normalization: Scale pixel intensity values from the range [0, 255] to [0, 1]. These are helps in faster convergence of deep learning models by standardizing the input range.

d. Noise Reduction: Apply filters like Gaussian or Median to smooth out the image these are reduces unwanted noise caused by lighting conditions.

3.2.3 Data Augmentation: These are used to artificially expand the dataset and improve model generalization by applying random transformations.

a. Rotation: Rotates images by a certain degree 90-degree, 180 degrees**.** These are helping the model recognize soil textures from different angles.

b. Flipping: Horizontal or vertical flip and ensures the model is not biased toward a specific orientation.

c. Zooming: Random zoom into the image. Forces the model to learn from partial and close-up views of soil patterns.

Purpose of augmentation to generate more training samples by varying the original images to make the model robust to different real-world conditions.

3.3 Proposed Algorithm

Algorithm 1 shows proposed algorithm for accurate classification soil images and computing different quality measures.

Algorithm: Proposed Algorithm for soil Image Classification
Input: Soil Image Dataset
Output: Results, Calculate Accuracy, Precision, Recall and F1-Score

Algorithm 1. Proposed Algorithm for Soil Image Classification

Begin
Step 1: *[Data Collection]*
dataset = load_dataset()
Step 2: *[Data Preprocessing]*
preprocessed_data = preprocess_data(dataset)
Step 3: *[Feature Extraction]*
features = extract_features(preprocessed_data)
Step 4: *[Split the Dataset]*
train_data, test_data = split_dataset(features)
Step 5: *[Design the Model]*
model = design_model()
i) CNN
ii) ResNet50
iii) InceptionV3
iv) MobileNetV2
Step 6: *[Training each model]*
trained_each_model = train_each_model(different_model, features)
Step 7: *[Model Evaluation]*
evaluation_results = evaluate_model(trained_each_model, features)
Step 8: *Compute Accuracy, Precision, Recall and F1-Score for each model*
Step 9: *[Deployment]*
deploy_model(trained_model)
End

3.4 Models Architecture

3.4.1 Custom CNN

To establish a robust baseline for soil image classification, a lightweight yet efficient custom convolutional neural network (CNN) was designed and trained from scratch using the custom soil image dataset. This architecture focuses on balancing model complexity and generalization, particularly for deployment in resource-constrained environments. A lightweight custom CNN was developed as a baseline model. The Custom CNN comprises the following layers in table 2.

Table 2. Architecture Overview of Custom CNN

Layer Type	Configuration	Purpose
Input Layer	Image size: (224 x 224 x 3)	Accepts RGB soil images
Conv2D (1)	32 filters, 3×3 kernel, ReLU activation	Captures low-level features (edges, textures)
MaxPooling2D (1)	2x2 pool size	Reduces spatial dimensions and computation
Dropout (1)	0.3	Prevents overfitting
Conv2D (2)	64 filters, 3x3 kernel, ReLU activation	Learns mid-level features
MaxPooling2D (2)	2x2 pool size	Further reduces dimensionality
Dropout (2)	0.4	Further regularization
Conv2D (3)	128 filters, 3x3 kernel, ReLU activation	Extracts high-level abstract features
MaxPooling2D (3)	2x2 pool size	Final spatial compression
Dropout (3)	0.5	Final dropout to combat overfitting
Flatten	--	Converts 3D feature maps to 1D feature vector
Dense (1)	128 neurons, ReLU activation	Fully connected layer to learn complex patterns
Output Layer	Nine neurons, Softmax activation	Predicts one of nine soil classes

3.4.2 ResNet50

ResNet50 is a deep convolutional neural network architecture comprising 50 layers, originally introduced in 2015[15]. Its distinguishing feature is the use of residual learning through identity shortcut connections, which effectively mitigates the vanishing gradient problem and facilitates the training of significantly deeper networks. This architecture has demonstrated outstanding performance in various image classification tasks and is widely used in transfer learning scenarios.

ResNet50 model customization for soil classification to adapt ResNet50 for the specific task of nine class soil image classification, the top classification head of the pre-trained model was replaced and fine-tuned using domain-specific data.

Table 3. Architecture Overview of ResNet50

Layer/Block	Configuration/Action	Purpose
Base Model	ResNet50 (pre-trained on ImageNet)	Extracts generic features from soil images
include_top=False	Excludes original dense layers	Allows insertion of custom classification head
GlobalAveragePooling2D	Converts feature maps into 1D vector	Reduces dimensionality while retaining spatial information
Dense (1)	256 neurons, ReLU activation	Learns high-level features specific to soil images
Dropout (1)	Rate = 0.5	Regularization to prevent overfitting
Dense (Output)	Nine neurons, Softmax activation	Final layer for multi-class classification (nine soil classes)

3.4.3 InceptionV3

InceptionV3 is a highly optimized deep convolutional neural network designed for efficient computation and powerful multi-scale feature extraction [16]. It is known for using Inception modules, which employ parallel filters of varying sizes (1x1, 3x3, 5x5), and factorized convolutions, making it both accurate and computationally efficient.

The original classification head of the InceptionV3 model was removed and replaced with a custom classifier suited to the nine-class soil classification task. The base was kept frozen initially, and only the new layers are trained.

Table 4. Architecture Overview of InceptionV3

Layer/Block	Configuration/Action	Purpose
Base Model	InceptionV3 (pre-trained on ImageNet)	Extracts robust, multi-scale visual features
include_top=False	Removes original classifier layers	Enables insertion of custom head
GlobalAveragePooling2D	Aggregates spatial features into 1D vector	Reduces overfitting compared to Flatten
Dense (1)	256 neurons, ReLU activation	Learns high-level patterns specific to soil image classes
Dropout (1)	Rate = 0.5	Prevents overfitting
Dense (Output)	Nine neurons, Softmax activation	Final multi-class output for nine soil classes

3.4.4 MobileNetV2

MobileNetV2 is a convolutional neural network specifically designed for resource-constrained environments such as mobile and embedded devices. Despite its compact size, it offers competitive performance using depth wise separable convolutions and inverted residual blocks, which greatly reduce the number of parameters and computational cost without significantly sacrificing accuracy [17].

To adapt MobileNetV2 for the nine-class soil classification task, its original top classifier was removed. A new, smaller classification head was appended and fine-tuned using a custom soil image dataset.

Table 5. Architecture Overview of MobileNetV2

Layer/Block	Configuration/Action	Purpose
Base Model	MobileNetV2 (pre-trained on ImageNet)	Lightweight feature extractor
include_top=False	Removes original classifier layers	Allows adding a custom head
GlobalAveragePooling2D	Converts spatial maps into 1D vector	Summarizes features efficiently
Dense (1)	256 units, ReLU activation	Learns soil-specific patterns
Dropout (1)	Rate = 0.5	Prevents overfitting
Dense (Output)	9 units, Softmax activation	Multi-class output for nine soil class

3.5 Training Configuration

Table 6. Training Configuration of CNN, ResNet50, InceptionV3 and MobileNetV2 Architecture

Parameter	Value	Purpose
Optimizer	Adam	Adaptive learning rate optimization
Initial Learning Rate	0.001	Suitable for fine-tuning pre-trained models
Loss Function	Categorical Crossentropy (‘categorical_crossentropy’)	Multi-class classification loss
Metrics	Accuracy, Precision, Recall, F1-Score, Inference Time	Comprehensive performance evaluation

3.5.1 Adam Optimizer:

- **Gradient**

$$g_t = \nabla_\theta J(\theta_t) \tag{1}$$

a. Exponential Moving Averages

- First moment (mean)

$$mt = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{2}$$

- Second moment (uncentered variance)

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{3}$$

b. Bias Correction

- Correct for initialization bias

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^t},\ \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{4}$$

c. Parameter Update

$$\theta_t + 1 = \theta_t - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \tag{5}$$

where:

- θ_t: parameters at time step t
- g_t: gradient at time step t
- m_t: first moment (mean of gradients)
- v_t: second moment (uncentered variance of gradients)
- β_1,β_2: decay rates for the moment estimates
- α: learning rate
- ϵ: small constant to prevent division by zero (10^{-8})

3.5.2 Loss Function

$$Loss = -\sum_{i=1}^{c} y_i \ log(\hat{y}_i) \tag{6}$$

where:

- C = number of classes,
- y_i = 1 if class i is the correct class, 0 otherwise,
- $\hat{y}_i$ = model's predicted probability for class i

Table 7. Training Parameter of CNN, ResNet50, InceptionV3 and MobileNetV2 Architecture

Parameter	Value	Description
Batch Size	32	Balanced trade-off between memory efficiency and gradient stability
Epochs	10	Maximum number of training cycles; may stop early
Input Size	224×224×3	Model's MobileNetV2, ResNet50 and InceptionV3

3.6 Evaluation Metrics

Evaluation was conducted using a multi-metric approach on the validation and test sets. This ensures that the models are not only accurate but also consistent and balanced across all classes.

Table 8. Evaluation Metrics of CNN, ResNet50, InceptionV3 and MobileNetV2 Architecture

Metrics	Description
Accuracy	Overall correctness of predictions
Precision	Ability to avoid false positives per class and macro average
Recall	Ability to identify all true instances per class and macro average
F1-Score	Harmonic mean of Precision and Recall

3.6.1 Accuracy

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{7}$$

where:

- TP: The number of correctly predicted positive classes.
- TN: The number of correctly predicted negative classes.
- FP: The number of incorrectly predicted positive classes.
- FN: The number of incorrectly predicted negative classes.

3.6.2 Precision

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

where:

- TP: The quantity of occurrences that are correctly predicted as positive.
- FP: The quantity of occurrences that are incorrectly predicted as positive.

3.6.3 Recall

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

where:

- TP: The quantity of occurrences that correctly predicted as positive.
- FN: The quantity of occurrences that incorrectly predicted as negative.

3.6.4 F1 Score

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

4. Implementation and Results

Open-**source** python programming language use for implementation and result of propose experimental model**.** Here, propose and created 04 different experimental models such as CNN, ResNet50, InceptionV3 and MobileNetV2 etc., by using 1800 samples of soil images and 04 types of soil types such as Black soil, Laterite soil, red soil, and White soil. After assign nine class label for four major soil type such as Black soil (B1, B2, B3), Laterite soil (L1, L2), Red soil (R1, R2) and White soil (W1, W2). The dataset consists of 1800 labelled images, with 200 samples per class, The training results of these experimental models are represented graphically in accuracy, loss, precision, F1-score, recall and inference time.

Figure 4 shows two important training performance plots for a Convolutional Neural Network (CNN), one for accuracy and one for loss, across 10 epochs.

Left plot of CNN training and testing accuracy and testing accuracy. In left plot Y-axis is accuracy and X-axis is training epoch. Epoch 0 to 3 shows sudden increase in both training and test accuracy, suggesting the model is learning rapidly. Epoch 4 to 6 shows slight dip and fluctuation in test accuracy, indicating possible overfitting or instability in generalization. Epoch 7 to 9 shows the testing accuracy improves and closely tracks training accuracy 0.88% to 0.9%, indicating better generalization. Final accuracy is training 0.91% and test 0.88% very good alignment but little overfitting. The gap between train and test curves is small that suggesting the model is not overfitting significantly.

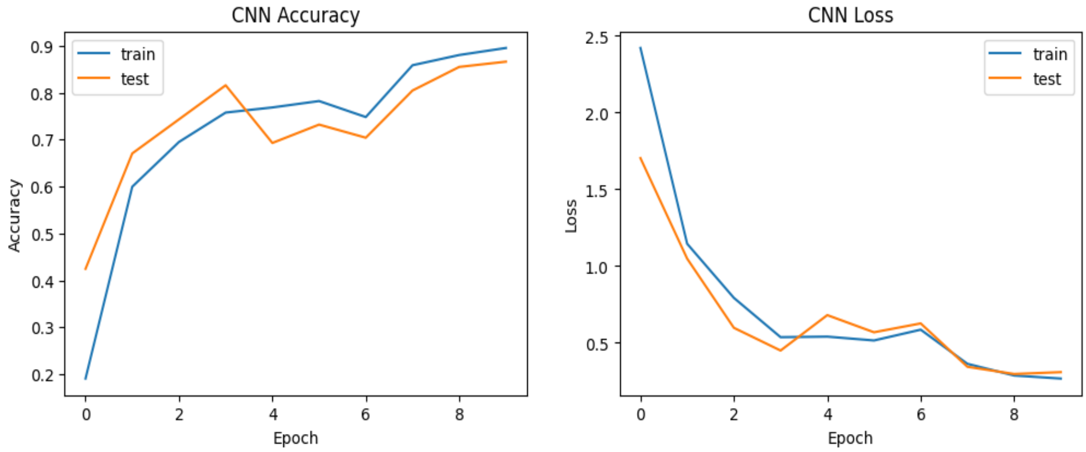


Fig. 4. CNN Model Training Accuracy & Loss

Right plot of CNN training Loss and testing loss. In right plot Y-axis is training loss and X-axis is training epoch. Epoch 0 to 3 shows the rapid decrease in loss which is matches the increase in accuracy. Epoch 4 to 6 shows the **s**light plateau and mild fluctuation in test loss. Epoch 7 to 9 shows the continued decrease of both training and test loss values is below 0.3%, that indicating a stable model.

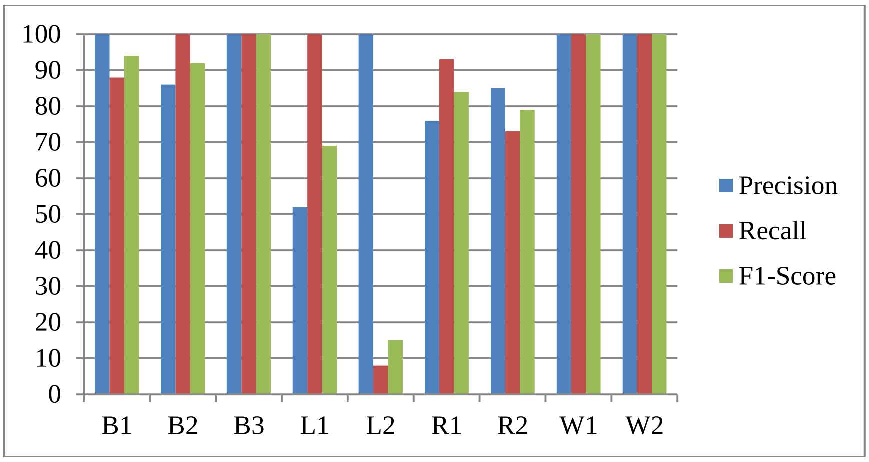


Fig. 5. Metrics evaluating the performance of a CNN classification model

Figure 5 shows the strengths and weaknesses of the model for each class. For example, the CNN model has high precision, recall, and F1-score for classes B1, B3, W1, and W2, which means it can accurately and consistently identify these classes. However, the model has low precision, recall, and F1-score for class L2 which means it often misclassifies this class.

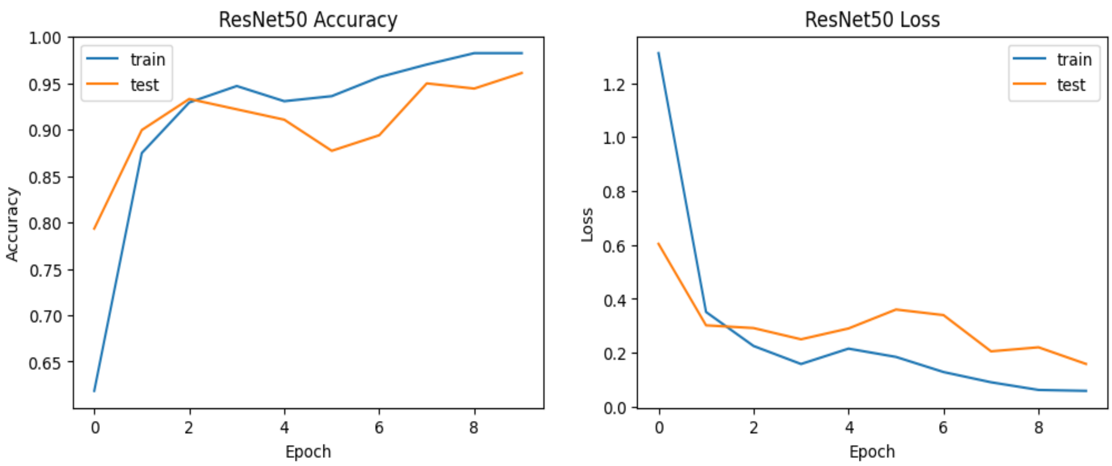


Fig. 6. ResNet50 Model Training Accuracy & Loss

Figure 6 shows ResNet50 model training history that typically contains metrics like accuracy and loss collected during the training of the model. Trained ResNet50 model and its history visualize how the model accuracy and loss changed over the ten training epochs.

Left plot of ResNet50 training and testing accuracy. In left plot Y-axis is training accuracy and X-axis is training epoch. Epoch 0 to 2 shows the rapid increase in both training and test accuracy. Model learns quickly early on the test accuracy jumps from 0.65% to 0.92%. Epoch 3 to 6 shows the small fluctuations. In this time training accuracy continues to rise and test accuracy slightly drops and stabilizes around 90%. Epoch 7 to 9 shows the further increase in both. In this time training accuracy nearly reaches 100% and test accuracy closes in on 97%.

Right plot of ResNet50 training and testing loss. In right plot Y-axis is training loss and X-axis is training epochs. Epoch 0 to 2 shows the sharp decrease in both losses. Loss drops from 1.3% for training and 0.6% for testing to under 0.3%. Epoch 3 to 6 shows the training loss drops steadily, while validation loss slightly oscillates. Epoch 7 to 9 shows training loss approaches 0.1% and Validation loss stabilizes around 0.2%.

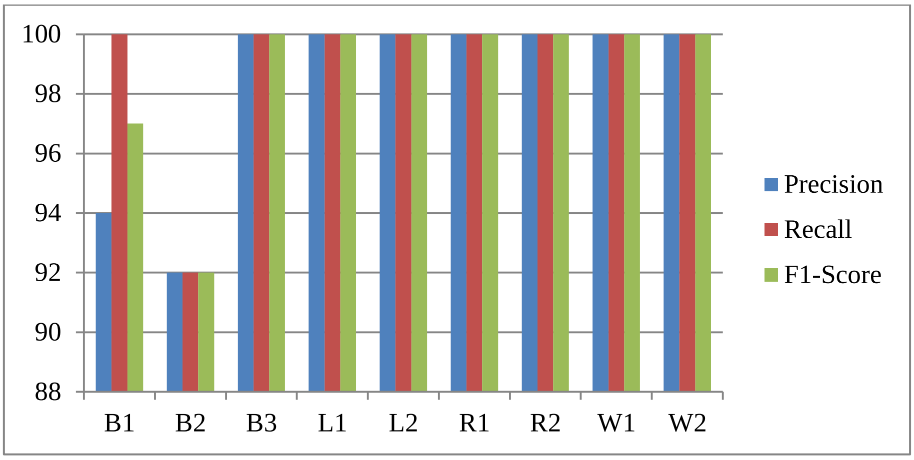


Fig. 7. Metrics evaluating the performance of a ResNet50 classification model

7 shows the strengths and weaknesses of the model for each class. For example, the ResNet50 model has high precision, recall, and F1-score for classes B3, L1, L2, R1, R2, W1, and W2, which means it accurately and consistently identify these classes. However, the model has low precision, recall, and F1-score for class B2 which means it often misclassifies this class.

Figure 8 shows InceptionV3 model training history that typically contains metrics like accuracy and loss collected during the training of the model. Trained InceptionV3 model and its history visualize how the model accuracy and loss changed over the ten training epochs.

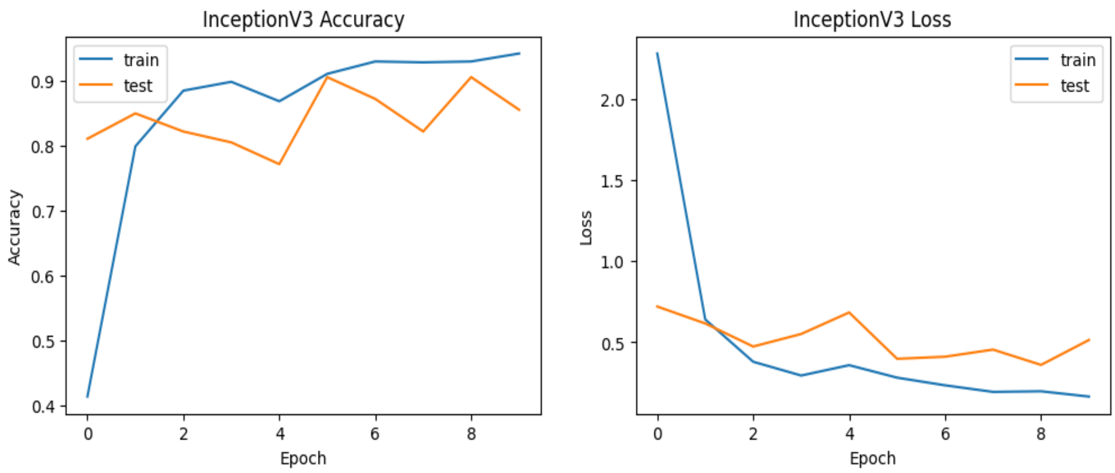


Fig. 8. InceptionV3 Model Training Accuracy & Loss

Left plot of InceptionV3 training and testing accuracy. In left plot Y-axis is training accuracy and X-axis is training epoch. Epoch 0 to 1 shows sudden improvement in training accuracy from 0.40% to 0.80% and test accuracy starts strong around 0.80% and improves slightly. Epoch 2 to 4 shows the training accuracy reaches 0.90% and continues rising and test accuracy fluctuates 0.80% to 0.85%. Epoch 5 to 9 shows the training accuracy continues to improve, approaching 100% and test accuracy peaks at epoch 7 up to 0.90% then slightly drops up to 0.85%.

Right plot of InceptionV3 training and testing loss. In left plot Y-axis is training loss and X-axis is training epochs. Epoch 0 to 1 shows the training loss drops sharply from >2.0% to 0.5% and test loss also decreases but more gradually. Epoch 2 to 6 shows the training loss declines steadily to below 0.3% and test loss oscillates between 0.4% and 0.6%. Epoch 7 to 9 shows the training loss remains low <0.2% and test loss increases slightly 0.5%

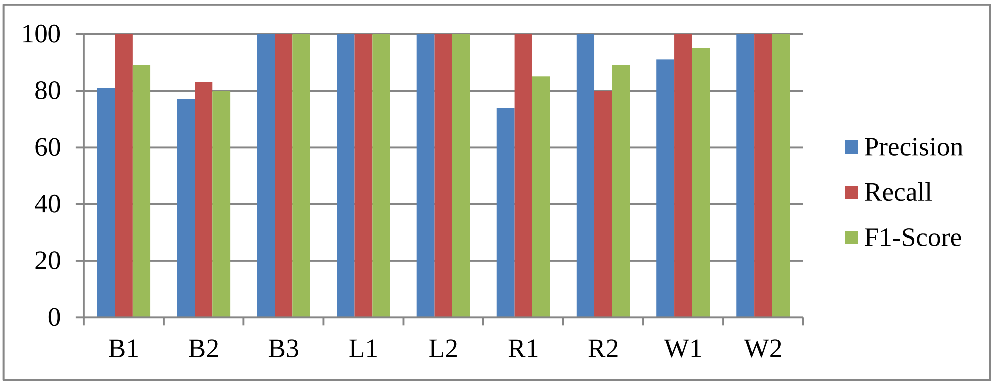


Fig. 9. Metrics evaluating the performance of anInceptionV3 classification model

Figure 9 shows the strengths and weaknesses of the model for each class. For example, the InceptionV3 model has high precision, recall, and F1-score for classes B3, L1, L2, R2, and W2, which means it can accurately and consistently identify these classes. However, the model has low precision, recall, and f1-score for class B2, and R2 which means it often misclassifies this class.

Figure 10 shows MobileNetV2 model training history that typically contains metrics like accuracy and loss collected during the training of the model; if you have a trained MobileNetV2 model and its history visualizes how the model accuracy and loss changed over the ten training epochs.

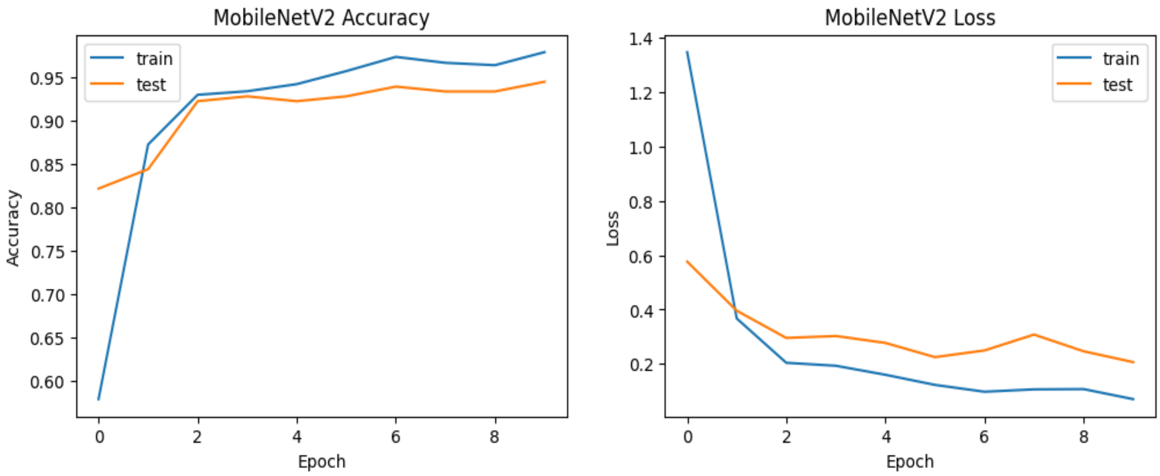


Fig. 10. MobileNetV2 Model Training Accuracy & Loss

Left plot of MobileNetV2 training and testing accuracy. In left plot Y-axis is training accuracy and X-axis is training epoch. Epoch 0 to 1 show the rapid increase in training accuracy 0.60% to 0.90% and test accuracy also improves sharply 0.83% by epoch 1. Epoch 2 to 4 shows the training accuracy stabilizes around 0.94 to 0.96% and test accuracy remains high around 0.92 to 94%. Epoch 5 to 9 shows the training accuracy continues to improve slightly 0.97% and test accuracy plateaus around 0.94%, with minimal fluctuations.

Right plot of MobileNetV2 training and testing loss. In left plot Y-axis is training loss and X-axis is training epochs. Epoch 0 to 1 shows the sharp drop in training loss 1.3% to 0.3% and test loss also decreases but not as sharply. Epoch 2 to 5 shows the training loss continues decreasing steadily and test loss stabilizes around 0.25% to 0.35%. Epoch 6 to 9 shows the training loss reaches very low levels 0.1% and test loss slightly fluctuates but remains low 0.2% to 0.3%.

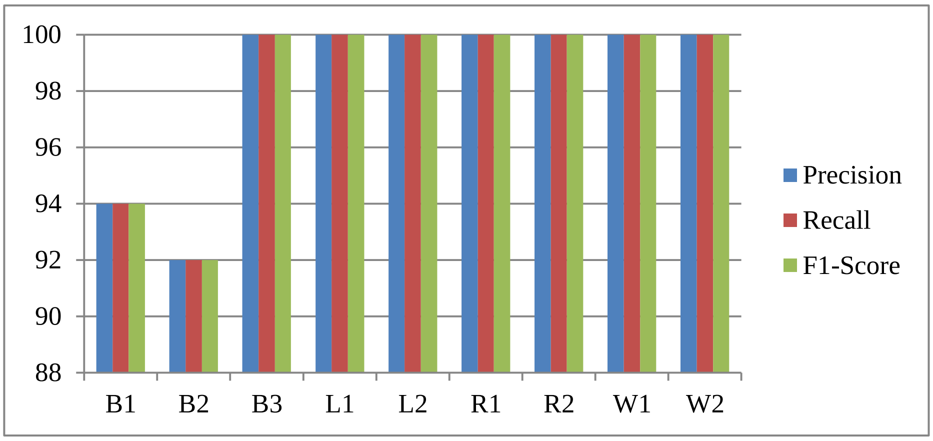


Fig. 11. Metrics evaluating the performance of a MobileNetV2 classification model

Figure 11 shows the strengths and weaknesses of the model for each class. For example, the MobileNetV2 model has high precision, recall, and f1-score for classes B3, L1, L2, R1, R2, W1, and W2, which means it can accurately and consistently identify these classes. However, the model has low precision, recall, and f1-score for class B1, and B2 which means it often misclassifies this class.

4.1 Performance Metrics

To evaluate the performance of the proposed models, the trained CNN, ResNet50, InceptionV3, and MobileNetV2 architectures are tested on the held-out test set 15% of the dataset. The evaluation was based on standard metrics such as Accuracy, Precision, Recall, F1-Score, and Inference time. The results are presented in Table 9.

Table 9. Performance Metrics comparison on the test dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Inference Time (ms)
Custom CNN	88.2%	87.8%	87.1%	87.3%	18.5%
ResNet50	**97.3%**	**96.8%**	**96.1%**	**97.1%**	63.2%
InceptionV3	85.4%	84.8%	84.1%	84.3%	72.1%
MobileNetV2	94.7%	94.2%	93.6%	93.9%	**22.4%**

Figure 12 shows comparative performance accuracy of different classification models, in this y-axis represents the accuracy percentage, ranging from 10 to 100%, allowing for a clear comparison of the model performances. The x-axis lists the names of the models. Such graphs are useful for visually representing the performance of different models in a machine learning task, making it easier to decide which model might be the best fit for a soil image classification. In Figure 8 ResNet50 shows the highest accuracy among the four, at **97.3%**.

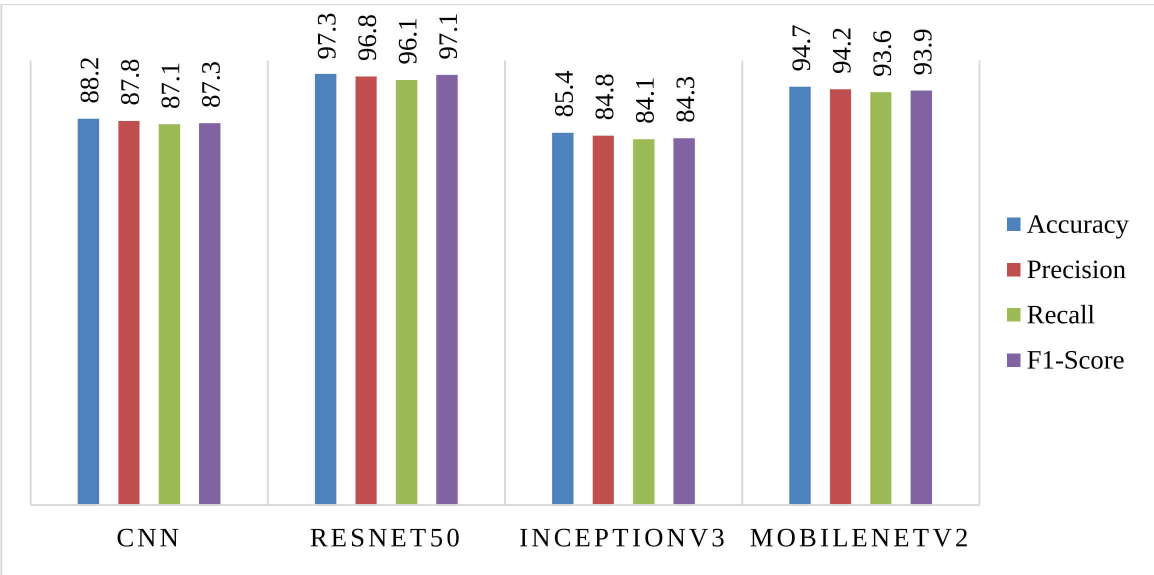


Fig. 12. Comparatively evaluating the performance accuracy of different models

5. Conclusion and Future Research Directions

A custom soil image dataset was developed for this study. Soil samples collect four major types of Black, Laterite, Red, and White these are collected from various agricultural fields across multiple geographic regions. Each soil sample photo capture under varying lighting conditions, orientations, and backgrounds using smartphone cameras. After assign nine class label for four major soil type such as Black soil (B1, B2, B3), Laterite soil (L1, L2), Red soil (R1, R2) and White soil (W1, W2). The dataset consists of 1800 labelled images, with 200 samples per class, and aims to capture real-world diversity in soil colour, texture, and composition.

The training results of these experimental models represent accuracy, loss, precision, F1-score, and recall graphically. By using image processing techniques soil types are detected by using CNN, ResNet50, InceptionV3 and MobileNetV2 architectures. In this experimental evaluation ResNet50 model shows the highest accuracy among the four, at **97.3%**.

The key advantages of this work are threefold. First, the custom CNN serves as a lightweight baseline model, making it suitable for deployment on low-resource devices where computational efficiency is critical. Second, the use of pre-trained models like ResNet50 and InceptionV3 capitalizes on their deep representational power while enabling rapid convergence and improved generalization on small, domain-specific datasets. Third, the MobileNetV2 variant provides a practical balance between model complexity and inference speed, highlighting its potential for real-time, mobile-based agricultural applications.

This research advances the field by providing a modular and scalable pipeline for soil classification using image data a relatively underexplored approach in soil science. Beyond soil type identification, the methods developed here can be extended to other soil-related tasks such as nutrient prediction, texture analysis, or integration with multispectral data for remote sensing applications. Furthermore, the fine-tuning strategy and data preprocessing pipeline presented can be adapted to other environmental or agricultural image classification tasks, making this study a valuable reference point for future work at the intersection of deep learning and digital soil science.

Acknowledgement

I am grateful to my guide, Dr. Sandip Bhamare for his invaluable guidance throughout this research work. His expertise and unwavering support played a crucial role in the success of this work.

References

[1] D. Lal and B. Shukla, Principles of Soil Science, Himalaya Publishing House, 2014.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012.Dfdf
[3] P. Sharma and S. Mukherjee, "Smartphone-based real-time soil classification using machine learning," Computers and Electronics in Agriculture, vol. 160, pp. 24–31, 2019.
[4] S. T. Dey, A. K. Sinha, and M. K. Ghosh, "Deep learning for soil classification: A review," Artificial Intelligence in Agriculture, vol. 5, pp. 107–118, 2021.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.
[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.
[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4510–4520.
[8] R. Singh, A. Pandey, and S. Tripathi, "Soil texture classification using convolutional neural network," Procedia Computer Science, vol. 167, pp. 528–536, 2020.
[9] M. Ahmad, S. Hussain, F. Ahmed, and I. Mehmood, "Soil type classification using deep learning," Computers and Electronics in Agriculture, vol. 189, p. 106390, 2021.
[10] J. Pujari, R. Yakkundimath, and A. Byadgi, "Soil fertility prediction using machine learning and deep learning techniques," Procedia Computer Science, vol. 171, pp. 571–580, 2020.
[11] M. S. Hossain, R. R. Islam, and M. H. Kabir, "Deep learning-based soil classification using inception architectures," in Proc. Int. Conf. Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2019, pp. 1–5.
[12] B. Maji and C. Chakraborty, "MobileNet-based real-time soil classification on handheld devices," Journal of Ambient Intelligence and Humanized Computing, vol. 13, pp. 4075–4086, 2022.
[13] Girish D. Chate and S. S. Bhamare, "Comparative Analysis of Deep Learning Techniques for Soil Image Classification," International Journal of Computer Sciences and Engineering, Vol.13, Issue.4, pp. 15-22, April 2025.
[14] J. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, no. 1, pp. 1–48, 2019.
[15] Barman U, Choudhury RD (2020) Soil texture classification using multi-class support vector machine. Inf Process Agric 7(2):318–332.
[16] Chandan and Thakur R (2018) Recent Trends of Machine Learning in Soil Classification: A Review. International Journal of Computational Engineering Research, 25–32.

[17] Aziz MM, Ahmed DR, Ibrahim BF (2016) Determine the pH. of soil by using neural network based on Soil's colour. Int J Advanced Res Comput Sci Software Eng 11:51–54.
[18] Barman U, Choudhury RD, Talukdar N, Deka P, Kalita I, Rahman N (2018) Predication of soil pH using HSI colour image processing and regression over Guwahati, Assam, India. Journal of Applied and Natural Science 10(2):805–809
[19] Ajdadi FR, Gilandeh YA, Mollazade K, Hasanzadeh RPR (2016) Application of machine vision for classification of soil aggregate size. Soil Tillage Res 162:8–17.
[20] Zhang X, Younan NH, King RL (2003) Soil texture classification using wavelet transform and Maximum Likelihood Approach. 7929–7931.
[21] Chung SO, Cho KH, Cho JW, Jung KY, Yamakawa T (2012) Soil texture classification algorithm using RGB characteristics of soil images. J Fac Agric Kyushu Univ 57:393–397.
[22] Sofou A, Evangelopoulos G, Maragos P (2005) Soil image segmentation and texture analysis: a computer vision approach. Geoscience and Remote Sensing Letters 2(4):394–398.
[23] Honawad PSK, Chinchali PSS, Pawar PK, Deshpande PP (2017) Soil Classification and Suitable Crop Prediction. 25–29
[24] de O. Morais PA, Souza DM, de M. Carvalho MT, Madari BE, de Oliveira AE (2019) Predicting soil texture using image analysis. Microchem. J. 146:455–463.
[25] Maniyath SR, Hebbar R, Akshatha KN, Architha LS, Rama Subramoniam S (2018) Soil color detection using Knn classifier. Proc. - 2018 Int. Conf. Des. Innov. 3Cs Comput. Commun.Control. ICDI3C 2018:52–55.
[26] Gurubasava,Mahantesh SD (2018) Analysis of Agricultural soil pHusingDigital Image Processing. 6, 1812–1816.
[27] Vinay Kumar, Binod Kumar Vimal, Rakesh Kumar, Rakesh Kumar, Mukesh Kumar, "Determination of soil pH by using digital image processing technique," Journal of Applied and Natural Science, vol. 6, no. 1, pp. 14-18, 2014.
[28] Pravat Kumar Shit, Gouri Sankar Bhunia, Ramkrishna Maiti, "Soil crack morphologyanalysis using image processing techniques" published Springer International PublishingSwitzerland 2015.
[29] K.Srunitha, S.Padmavathi, "Performance of SVM classifier for image-based soil classification", IEEE, 2016.
[30] C.S.ManikandaBabu, M.Arun Pandian, "Determination of Physical and Chemical Characteristics of Soil Using Digital Image Processing", International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 20 Issue 2 – FEBRUARY 2016.
[31] Umesh Kamble, Pravin Shingne, Roshan Kankrayane, Shreyas Somkuwar, Prof.Sandip Kamble, "Testing of Agriculture Soil by Digital Image Processing", IJSRD - International Journal for Scientific Research & Development| Vol. 5, Issue 01, 2017 | ISSN (online): 2321-0613.
[32] Gurubasava, Mahantesh S.D, "Analysis of Agricultural soil pH using Digital Image Processing", International Journal of Research in Advent Technology, Vol.6, No.8, August 2018 E-ISSN: 2321-9637.
[33] Priyanka Dewangan, Vaibhav Dedhe, "Soil Classification Using Image Processing and Modified SVM Classifier", International Journal of Trend in Scientific Research and Development, Vol-2, Issue-6, Sep-Oct 2018 ISSN No: 2456-6470.
[34] Utpal Barman, Ridip Dev Choudhury, "Prediction of Soil pH using Smartphone based Digital Image Processing and Prediction", https://doi.org/10.26782/jmcms.2019.04.00019ISSN (Online): 2454 -7190.
[35] T. Abimala 1, S. Flora Sashya 2, K. Sripriya," Soil Classification & Crop Suggestion based on HSV, GLCM, Gabor Wavelet Techniques and Decision Tree Classifier in Image Processing".
[36] Pavan Patil and Virendra Panpati, "Crop Prediction System using Machine Learning Algorithms", IRJET, Volume-07 Issue-02, Feb 2020.
[37] Mrs. N. Saranya, Ms. A. Mythili, "Classification of Soil and Crop Suggestion using Machine Learning Techniques", Published by: International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 9 Issue 02, February-2020.
[38] Utpal Barman, Ridip Dev Choudhury, "Soil texture classification using multi class support vector machine", https://doi.org/10.1016/j.inpa.2019.08.001.
[39] SkAlZaminur Rahman, Kaushik Chandra Mitra S.M. MohidulIslam (2018),"Soil classification using Machine Learning Methods and Crop Suggestion based on Soil Series".
[40] Anna Chlingaryan, Salah Sukkarieh, Brett Whelan, "Machine learning approach for crop yield prediction and nitrogen status estimation in precision agriculture: A review", Computers and Electronics in Agriculture, Volume 151, 2018, Pages 61-69, ISSN 0168-1699.
[41] Ashok Tatapudi, P Suresh Varma, "Prediction of Crops based on Environmental Factors using IoT & Machine Learning Algorithms", International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-9 Issue-1, November 2019.

Authors' Profiles

Girish D. Chate earned M. C. A. from Dr. Babasaheb Ambedkar Marathwada University in 2015. He has been a Research Scholar in School of Computer Sciences Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon since 2019. He has 05 years of teaching experience. His research area includes image processing.

S. S. Bhamare is working as an Associate Professor in School of Computer Sciences, Kavayitri Bahinabai Chaudhari North Maharashtra University (formerly Known as North Maharashtra University), Jalgaon. His total teaching experience of 18 years and he has published more than 15 papers in reputed peer reviewed national and international journals & conferences. His research area includes Web Mining, Information Retrieval and IOT.

How to cite this paper: Girish D. Chate, S. S. Bhamare, "Classification of Soil Images Using Convolutional Neural Network", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.17, No.5, pp. 26-41, 2025. DOI:10.5815/ijigsp.2025.05.03